Active-Rx: Mobile Activity Sensing on Smartphones

September 25, 2019

Abhinav R. Bandari and Anshita Saini
Advised by Parker S. Ruth
Ubiquitous Computing Laboratory
Paul G. Allen School of Computer Science and Engineering
University of Washington-Seattle

Abstract

A person's physical activity level is a vital sign of his or her health. Thus, an automatic exercise tracking application would be useful to doctors in monitoring their patients' activity level. We sought to explore activity monitoring in the smartphone through two avenues: 1) sonic motion sensing by measuring doppler shift 2) analysis of the phone's accelerometer and gyroscope data. We built a custom data collection tool to collect data on arm lifts, bicep curls, chair sit-to-stands, and non-exercise (control). We produced spectrograms from the audio data for each exercise. We then implemented image classification on these spectrograms using a multilayer perceptron model to distinguish between the different exercises. Ultimately, the classifier identified the correct activity with 95% accuracy. Additionally, we implemented a distinct multilayer perceptron model on the raw accelerometer and gyroscope data. Overall, our work demonstrates that characterization of the induced doppler shift with machine learning can be used to distinguish between similar physical motions.

1 Introduction

1.1 Rationale

The medical community generally maintains that there are six vital signs that indicate the status of an individual's health: body temperature, pulse, blood pressure, respiration, height, weight. Recently, physical activity has been proposed as an additional vital sign [1]. Furthermore, they introduced the notion of doctors prescribing exercise as a medicine. In addition, the rising obesity epidemic in the United States has motivated the need for increased physical activity. To that end, many tech companies are creating wearable devices to monitor physical activity, including but not limited to the FitBit, Apple Watch, and StepWatch. These devices measure step counts and traveled distance, and some can estimate calorie expenditure. However, these devices remain prohibitively expensive for the populations with the highest need for this technology. Furthermore, the aforementioned activity metrics do not adequately track stationary exercises commonly used by obese, disabled, and elderly individuals. Our work aims to improve upon these limitations of existing wearable activity trackers by developing an automatic exercise activity tracker on a commodity device that many users already own: the smartphone.

1.2 Related Work

One potential approach to activity monitoring is a classification system based on computer vision. The smartphone's camera can record the user exercising and video recognition algorithms can be used to distinguish between different exercises. For instance, Khurana et al. developed GymCam, which utilizes video recognition software to identify exercises from video footage at a gym [2]. They deployed their algorithms on a Logitech C922, a compute-intensive recording device. However, state-of-the-art video recognition algorithms are still too computationally expensive to deploy on smartphones, and smartphone camera recordings raise issues of privacy infringement. For this reason, we decided to not explore activity monitoring with computer vision.

Another approach to activity monitoring is sonic gesture recognition [3]. In this modality, the smartphone plays a tone at a constant frequency while the microphone records simultaneously. If the user makes any motion within approximately 1 meter of the device, the microphone records a frequency slightly higher or lower than the carrier frequency. This phenomenon is known as doppler shift. Different types of exercise may produce a distinct patterns of doppler shift. A machine learning classification model or even a heuristic algorithm can be utilized to distinguish between the doppler shift patterns of different exercises. Fu et. al developed a convolutional neural network to analyze the doppler shift patterns in spectrogram images and identify the exercise performed [4]. Their model can successfully identify the exercises toe touches, squats, and bicycle.

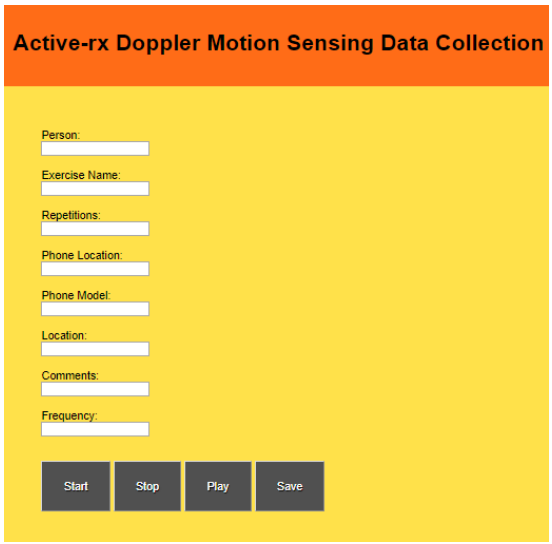
A third approach to activity monitoring is utilizing the smartphone's built-in accelerometer and gyroscope to characterize different types of motion. For example, Microsoft Research developed RecoFit, a wristband with an accelerometer and gyroscope, that utilizes machine learning to distinguish between exercises such as situps and pullups [5]. A smartphone-based system would require the user to hold their phone while exercising, posing a substantial limitation. However, we decided to explore this approach due to the high accuracy of RecoFit and other accelerometer-based motion classifiers. Furthermore, utilizing the smartphone's accelerometer and gyroscope provides a straightforward, accurate method of counting repetitions of the exercises.

2 Methods

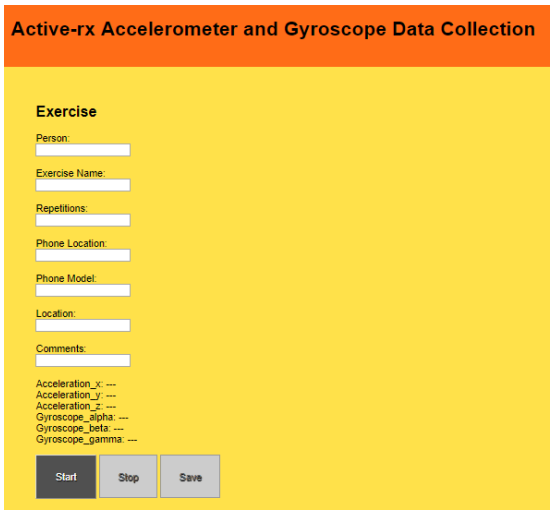
We explored two approaches to automated activity classification systems: 1) sonic doppler motion sensing and recognition 2) classification of accelerometer and gyroscope data. We decided to explore both of these paths by developing web

Activity	Half Repetition
Arm lift	Raise both arms simultaneously in the body's frontal plane until hands touch each other above head
Bicep curl	Raise both forearms simultaneously while keeping elbow stationary until fists touch shoulders
Chair sit-to-stand	Stand up from a seated position while extending both arms in front of the body

Table 1: Our definitions of each activity



(a) Audio data collection tool.



(b) Accelerometer and gyroscope data collection tool.

Figure 1: Our custom web data collection interfaces. The definitions of each text field are indicated in Table 2.

applications so that we could easily test and deploy our application across devices. To this end, we developed web data collection tools for both audio data (Figure 1a) and accelerometer/gyroscope data (Figure 1b).

2.1 Choice of Exercise Activities

Past activity recognition applications are fairly proficient in identifying traditional calisthenic and strength training exercises [2, 4, 5], such as pushups, squats, or situps. In response to this need we prototyped an activity recognition algorithm focusing on exercises that sedentary patients can perform. After conferring with clinicians from the University of Washington Medical Center, we decided to collect data for the following activities:

1. armlifts
2. bicep curls
3. chair sit-to-stands
4. non-exercise (control)

A few sedentary patients may find chair sit-to-stands difficult, yet this is considered a more viable exercise for this target audience than situps or pushups. A full repetition of each of the first three exercises enumerated above comprises the half repetition defined in Table 1 and the reversal of the half repetition. Non-exercise serves as a control to the three exercise activities in Table 1 and has no intrinsic periodicity. We define non-exercise as no motion at all or any natural motion a sedentary person would perform (e.g., scratching chin, crossing arms, gentle rocking).

2.2 Data Collection

We developed custom web data collection tools to record audio data (Figure 1a) as well as accelerometer and gyroscope data (Figure 1b). Audio was recorded in a single input channel and was sampled at a frequency of 48000 Hz. Accelerometer and gyroscope data was sampled at a frequency of approximately 60 Hz due to limitations of the JavaScript Web Audio API.

Text field	Definition
Person	name of data *collector*
Exercise Name	either "arm lift", "bicep curl", "chair stand", *or*"*nonexercise*" (case-insensitive and whitespace-insensitive)
Repetitions	number of repetitions of the *exercise*
Phone Location	both the location and orientation of the *smartphone* (eg. "upright on table")
Phone Model	Make and model of the smartphone (eg. *GooglePixel*)
Location	Physical location of data collection (eg. annex, $6thfloor$)
Comments	Miscellaneous notes (eg., "far from phone", "*nearphone*")
Frequency	Carrier Frequency in Hertz (frequency of *thetone*), defaults to 10,000 Hz

Table 2: Definition of each text field in the web data collection tool.

During data collection, the subject first recorded metadata information, e.g. name, activity type, number of repetitions, etc. (Table 2). The subject would then press the "Start" button, place the phone in the location and orientation defined in the text field "Phone Location" (Table 2), perform the exercise, and press the "Stop" button. We used a Google Pixel smartphone to collect our data. Due to the position of the microphone on the back of this particular model, we empirically determined that only motion behind the phone consistently created observable doppler shifts. Hence, although we intended to collect data with several different phone positions and orientations, all audio data was collected with the phone propped upright and facing away from the user (Figure 2). For the accelerometer and gyroscope data, the phone was held in the user's right or left hand, a further aspect added to the recorded information. The user then pressed the Start button, placed the phone in his or her hand in any gyrometric orientation and performed the number of repetitions recorded earlier.

Figure 2: Data collection setup with Google Pixel from the data collector's perspective.

In addition, we later cropped a fixed length of data off the start and end of each data stream to eliminate data associated with the user's motion when pressing the "Start" or "Stop" buttons. Even after data collection is halted, the data collector can edit the information in any of the text fields if necessary. If the data collector is confident that the data is clean and labelled correctly, they may press the "Save" button, which saves the recorded data to a server.

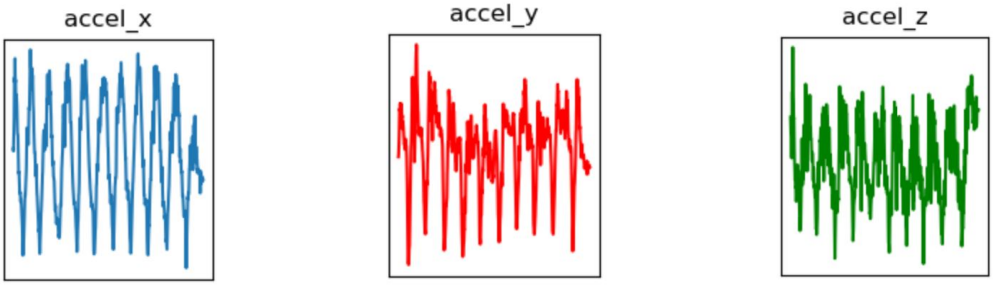
Using the above procedure, we collected several raw audio files of various exercises (Table 3). Each file contains data representing approximately 10 to 15 seconds of exercise activity. We later segment this data into smaller lengths. We additionally collected raw accelerometer and gyroscope data of two exercises, arm lifts and chair sit-to-stands (Table 3).

Type of Exercise	Audio Data Points Before Augmentation	Audio Data Points After Augmentation	Accelerometer Data Points
Arm lift	44	140	46
Bicep curl	55	153	-
Chair sit-to-stand	46	121	49
Non-exercise	26	157	-

Table 3: Number of instances of clean, labelled audio data for each class

2.3 Digital Signal Processing

We loaded the audio data from the server, processed the audio data in Python using the `SciPy`, `NumPy`, and `Matplotlib` libraries. We cropped 3 seconds (144,000 samples) off of the start and end of each audio signal to eliminate data associated with the subject's motion when starting and stopping data collection on the web interface. Then, we produced segments of length 2 seconds (96,000 samples) from the resulting signal using a sliding window with step size 48,000 samples. We chose this segment length because it was long enough to capture at least one full repetition of each exercise. Our rationale was that our machine learning model should be able to identify each exercise from each

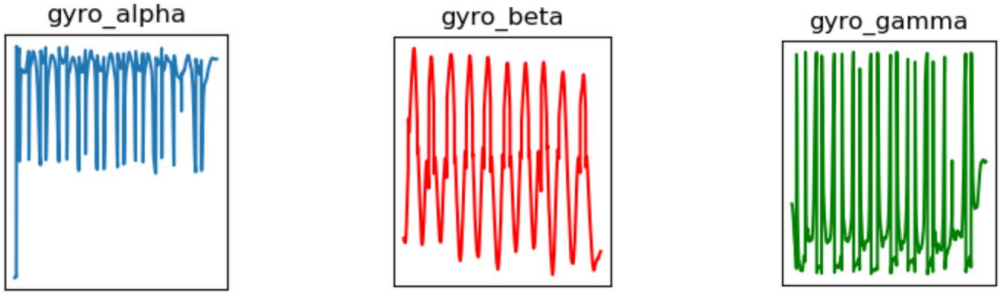


(a) $\textbf{accel}_{\textbf{x}}$: Component of the device's acceleration vector along the axis parallel to the top and bottom of the phone (x-axis).

(b) $\textbf{accel}_{\textbf{y}}$: The component of the device's acceleration vector along the axis parallel to the sides of the phone (y-axis).

(c) $\textbf{accel}_{\textbf{z}}$: The component of the device's acceleration vector along the axis normal to the plane containing the phone (z-axis).

Figure 3: Graph plots of phone's acceleration in three axes while user performs arm lift.



(a) $\textbf{gyro}_{\textbf{alpha}}$: The component of the device's angular displacement about the phone's z-axis.

(b) $\textbf{gyro}_{\textbf{beta}}$: The component of the device's angular displacement about the phone's x-axis.

(c) $\textbf{gyro}_{\textbf{gamma}}$: The component of the device's angular displacement about the phone's y-axis.

Figure 4: Graph plots of phone's angular position about three axes while user performs arm lift.

segment regardless of whether the segment starts from the beginning, middle, or end of an exercise. In addition, the sliding window approach simultaneously produced more data to train our model (Table 3).

We generated spectrograms on each of these shorter segments. We used 20,000-point Discrete Fourier Transforms (DFT), overlap of 19,000 points to increase temporal resolution, and Tukey window transforms with shape parameter 0.25 to minimize spectral leakage. The DFT was only computed from 41 frequency bins below the carrier frequency to 41 frequency bins above the carrier frequency. This range was chosen to best capture the doppler shift as a result of the user's motion. The spectrograms associated with each of the 4 activities can be seen in Figure 5.

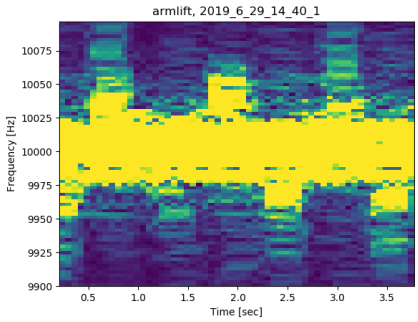
We approached data processing in a similar way for the accelerometer and gyroscope data by using the forenamed libraries. We encountered a challenge with processing the gyroscope data, as portions of the data above the range of Gyroscope interfaces were transposed to negative values. The derivative of the data was analyzed to find the parts of the data where this jump occurred, and these values were transposed to their positive counterparts. We then generated six different plots of the processed accelerometer and gyroscope data on the time domain – specifically, we plotted acceleration in the x, y, and z directions (Figure 3), as well as the gyroscope measurements around the z, x, and y axes of the phone (Figure 4). An infinite impulse response (IIR) bandpass filter with cutoff frequencies at 0.3 Hz and 3 Hz was applied on portions of the data to eliminate noise.

Fold	Accuracy
1	0.97
2	0.98
3	0.93
4	0.97
5	0.96
6	0.98
7	0.92
8	0.96
9	0.93
10	0.85
Average	**0.95**

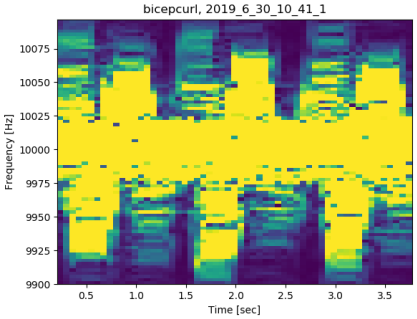
Table 4: Cross-Validation Scores

2.4 Machine Learning

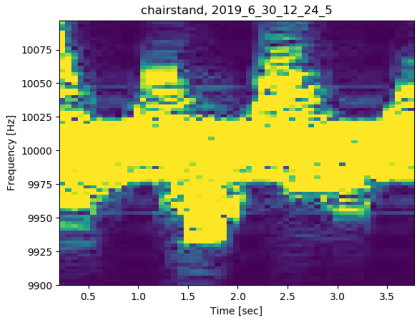
We then developed a 2D convolutional neural network (CNN), which received the 2-dimensional spectrogram array as input and outputted a 4x1 vector, where each value corresponds to the confidence score that the spectrogram belongs to one of the four classes listed in Table 1. The final prediction is the *argmax* of this output vector. The complete architecture of the CNN can be viewed in Figure 6.



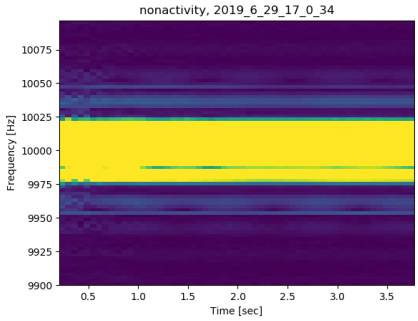
(a) Arm lift: Periodic alternation between negative doppler shift and positive doppler shift. As arm moves upward, it moves away from the phone, thereby inducing a negative doppler shift. As the arm moves downwards, it moves closer to the phone, thereby inducing a positive doppler shift.



(b) Bicep curl: During muscle contraction, as the angle between upper arm and forearm decreases from 180 to 90 degrees, forearm's motion towards the phone creates a small positive doppler shift. As the angle between the upper arm and forearm decreases from 90 to 0 degrees, forearm's motion away from the phone creates a large negative doppler shift. The inverse is observed during muscle relaxation.



(c) Chair sit-to-stand: As the user stands up, their whole body tends to move closer to the phone, thereby inducing a positive doppler shift. As the user sits down, their body moves away from the phone, inducing a negative doppler shift.



(d) Non-exercise: Minimal motion induces no observable doppler shift.

Figure 5: Spectrograms computed on audio signals associated with each of the 4 activities. Carrier frequency for these samples is 10,000 Hertz.

3 Results

We evaluated our model using 10-fold cross validation. No single fold contained segments of audio data produced from the same audio file, ensuring that cross-validation scores provided a fair estimate of the model's performance (Table 4). The final confusion matrix combined across all folds can be viewed in Table 5.

Following the aforementioned data processing, we used a threshold model to count repetitions of exercises. Lower and upper thresholds at 7.7 percent and 90 percent were applied to the data and a program iterated through arrays of the gyroscope data for arm lift exercises collected for 47 samples of either 1, 15, 16, or 20 repetitions. Repetitions were counted with an accuracy of 79.1%. We then constructed a 1D convolutional neural network for binary classification between the arm lift and chair sit-to-stand exercises.

	Arm lift	*Bicep curl*	*Chair sit-to-stand*	*Non-exercise*	*Total*
Arm lift	**129**	6	2	3	140
Bicep curl	2	**151**	0	0	153
Chair sit-to-stand	5	1	**107**	8	121
Non-exercise	0	0	1	**156**	157

Table 5: Results of spectrogram classification model

4 Conclusion and Discussion

The doppler shift patterns in the spectrograms were fairly unique to each of the three exercises we investigated. Thus, our classification model was able to identify the exercise with near-perfect accuracy. Our work therefore suggests motion sensing and classification through characterization of the induced doppler shift can be applied to a wide range of physical motions. In Fu et. al's system [4], the phone is positioned on the ground while the user performs the exercise over the phone. Their system distinguishes between toe touches, bicycles, and squats with great accuracy, but these exercises are fairly different. Our work demonstrates that analysis of doppler shift patterns can be used to distinguish more similar exercises like arm lifts and bicep curls as well.

Moreover, our threshold model demonstrates that accelerometer and gyroscope data can be used to somewhat reliably count repetitions of exercises. The graphs of the data reflect that the Javascript Sensor APIs are able to accurately distinguish between each repetition of the exercise, even with variations of how the exercise is performed. One limitation of implementing the accelerometer and gyroscope to distinguish between exercises and count repetitions is that the user must hold the phone, limiting the types of exercises the user can perform.

One limitation of the doppler shift classification model is that the convolutional neural network is fairly computationally intensive, creating significant latency when classifying exercise in real time. However, since the shape of each doppler shift pattern in the spectrogram is most important in identifying the type of exercise, perhaps we can decide on the most salient features of each spectrogram, compute them for each spectrogram, and pass these features as an input vector to a classical machine learning model (eg. support vector machine). We can potentially isolate the doppler shift associated with a single repetition in the spectrogram image through a flood fill algorithm and subsequently compute features like height, width, curvature, etc. With fewer input features to process, the machine learning inference model would run much faster in real time.

Another limitation is that the carrier frequency used for the motion sensing (10,000 Hz) is within the audible range for humans. Thus, repeated exposure to this sound could cause headaches or irritate the user. We used this frequency only because the microphones of the phone we used in this investigation (Google Pixel) did not register frequencies much higher than this. However, microphones on the latest iPhones or Android phones are all capable of registering ultrasound frequencies up to 22,000 Hz. Since humans cannot hear frequencies above 20,000 Hz, an activity sensing application using a carrier frequency between 20,000 Hz and 22,000 Hz would be more user-friendly.

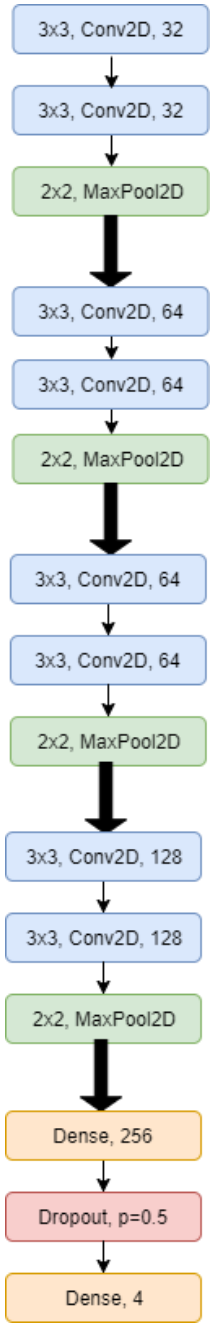


Figure 6: Architecture of the Convolutional Neural Network

References

[1] Michelle L Segar, Eva Guerin, Edward Phillips, and Michelle Fortier. From a Vital Sign to Vitality: Selling Exercise So Patients Want to Buy It. 15(4):6, 2016.

[2] Rushil Khurana, Karan Ahuja, Zac Yu, Jennifer Mankoff, Chris Harrison, and Mayank Goel. GymCam: Detecting, Recognizing and Tracking Simultaneous Exercises in Unconstrained Scenes. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(4):1–17, December 2018.

[3] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. SoundWave: Using the doppler effect to sense gestures. In *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems - CHI '12*, page 1911, Austin, Texas, USA, 2012. ACM Press.

[4] Biying Fu, Florian Kirchbuchner, Arjan Kuijper, Andreas Braun, and Dinesh Vaithyalingam Gangatharan. Fitness activity recognition on smartphones using doppler measurements. 15(4):14, 2018.

[5] Dan Morris, T. Scott Saponas, Andrew Guillory, and Ilya Kelner. RecoFit: Using a Wearable Sensor to Find, Recognize, and Count Repetitive Exercises. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 3225–3234, New York, NY, USA, 2014. ACM.